Secure Outsourcing of Key Update for Shared Data Critical Information Infrastructure

Snekha.S[1], Saranya.D[2], Sathea sree.S[3]

1. B.Tech Student, Dept of IT, Jerusalem College Of Engineering, Chennai, snekhait96@gmail.com
2. B.Tech Student, Dept of IT, Jerusalem College Of Engineering, Chennai, sharansanju96@gmail.com
3. Assistant Professor, Dept of IT, Jerusalem College Of Engineering, Chennai, satheasadasivam@gmail.com

Abstract –

Key-exposure problem has always been an important issue in many security applications. Recently, how to deal with this problem with regard of cloud storage has been proposed. The client stores their data in the cloud using the secret key. In the existing system, the client has to update the keys for each time period which is a burdensome task. In this design, all the key operations are done by Third Party Auditor (TPA). This design also equips the client with capability to further verify the validity of the encrypted secret keys. The Third Party Auditor (TPA) only needs to hold an encrypted version of the client's secret key while doing all these burdensome tasks on behalf of the client. The client requests the TPA and only needs to download the encrypted secret key from the Third Party Auditor (TPA). The goal of this paper is to design a cloud storage auditing protocol that can satisfy above requirements to achieve the outsourcing of key updates by using outsourcing and key update algorithms.

Index Terms—**Cloud storage auditing, outsourcing computation , key updating, Third Party Auditing.**

I. INTRODUCTION

Cloud storage is universally viewed as one of the most important services of cloud computing. Several types of cloud storage system have been developed supporting both personal and business users. The most basic form of cloud storage allows user to upload individual files or folders from their personal computers to central internet servers. This allows users to make backup copies of files incase their originals are lost. Users can also download their files from the cloud to other devices and also enable remote access to files for other people to share. The client store their massive data in the cloud storage. The cloud service provider (CSP) host the data of the data owners into the cloud which is accessed by data consumers. Although cloud storage provides great benefit to users, it brings new security challenging problems. People can now outsource time consuming computation workloads to cloud without spending the extra capital on deploying and maintaining hardware and software. Recently, outsourcing computation has attracted much and been researched widely. One of the most important security problem is how to efficiently check the integrity of the data stored in cloud. Now, many auditing protocols for cloud storage have been proposed to deal with this problem. These protocols focus on different aspects of cloud storage auditing such as the high efficiency , the privacy protection of data , the privacy protection of identities ,dynamic data operations , the data sharing , etc. The key exposure problem, as another important problem in recent years. Once the clients secret key is exposed to cloud , it faces many security problems. The Existing cloud storage auditing protocol with key-exposure resilience by updating the user's secret keys periodically. In this way, the damage of key exposure can be reduced. But it causes new local burdens for the client because the client has to execute the key update algorithm in each time period. Clients with limited computation resources such as mobile phones, they might hate doing such extra computations by themselves. It would be more attractive to make key updates as transparent as possible for the client. However, it needs to satisfy several new requirements to achieve this goal. At first, the real client's secret keys for cloud storage auditing should not be known by the authorized party who performs outsourcing computation for key updates. Else, it will bring the new security threat. So the authorized party should only hold an encrypted version of the user's secret key for cloud storage auditing. Secondly, the key updates should be completed under the encrypted state. In other words, this authorized party should be able to update secret keys for cloud storage auditing from the encrypted version he holds. Thirdly, it must be possible for the client to recover the real secret key from the encrypted version that is retrieved from the authorized party. Finally, the client should be able to verify the validity of the encrypted secret key after the client retrieves it from the authorized party. The aim of this paper is to design a cloud storage auditing protocol that can satisfy above requirements to achieve the outsourcing of key updates. The literature survey is discussed in section 2, the proposed idea is discussed in section 3, implementation and results are shown in section 4 and section 5 will sums up and conclude the paper.

II. LITERATURE SURVEY

Lot of research papers were studied and analyzed, few prototype solutions have been reported in past years and some of the recent research papers and descriptions are listed over here.

Jia Yu [1] et al 2016 describes about "Enabling cloud storage auditing with verifiable outsourcing of key updates". In this method all the key operations are done by Third Party Auditor (TPA) by reducing the burden of clients.

Cong Wang [2] et al 2013 describes about " Privacy preserving public auditing for secure outsourcing". This method guarantees the information respectability of the client. Here the clients depend on an autonomous outsider inspector (TPA) who has skill and proficient to review the outsourced information when required. In this method there is a threat for users data in distributed computing since the TPA performs audits for multiple users simultaneously.

Chris Erway [3] et al 2009 describes about "Dyna mic provable data possession". In this paper, TPA just needs to hold a scrambled variant of the customer's mystery key, while doing all these difficult assignments for the benefit of the customer. The customer just needs to download the scrambled mystery key from the TPA while transferring new documents to cloud. Moreover, our plan additionally outfits the customer with capacity to facilitate confirm the legitimacy of the scrambled mystery keys gave by TPA. We formalize the definition and the security model of this worldview. The security confirmation and the execution demonstrate that our point by point plan instantiations are secure and productive.

Giuseppe Ateniese [4] et al 2008 describes about "Scalable and efficient provable data possession". In this paper, they construct a highly efficient and provably secure PDP technique based entirely on symmetric key cryptography, while not requiring any bulk encryption where prior work is done using public key cryptography or requiring the client to outsource its data in encrypted form. Also, in contrast with its predecessors, our PDP technique allows outsourcing of dynamic data, i.e., it efficiently supports operations, such as block modification, deletion and append.

Rajalakshmi [5] et al 2014 describes about "An efficient audit service outsourcing for data integrity in clouds". In this paper, they address the construction of an interactive PDP protocol to prevent the fraudulence of prove and the leakage of verified data . They also prove that our construction holds these properties based on the computation Diffie– Hellman assumption and the rewind able black-box knowledge extractor. They also proposed an efficient mechanism with respect to probabilistic queries and periodic verification to reduce the audit costs per verification and implement abnormal detection timely. In addition, they present an efficient method for selecting an optimal parameter value to minimize computational overheads of cloud audit services.

III. PROPOSED SYSTEM

In this proposed system, key-update operations are performed by an authorized party(TPA) using AES key generation algorithm .The client downloads the encrypted secret key from the authorized party and decrypts to upload new files to cloud. This system includes the modules such as client authentication, upload data, auditing proof, key requirement and result. The architecture diagram of the proposed method is shown in Figure 1.

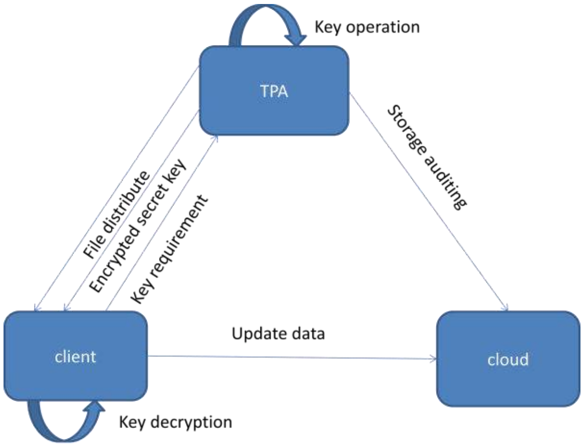


Figure 1. Architecture diagram.

The proposed system involves the following modules.

A. CLIENT AUTHENTICATION

The client is the owner of the files that are uploaded to cloud. The total size of these file is not fixed, the client can upload the growing files to cloud in different time points. The cloud stores the client's files and provides download service for the client.

After that, the client decrypts it to get his real secret key, generates authenticators for files, and uploads these files along with authenticators to cloud. In addition, the TPA will audit whether the files in cloud are stored correctly by a challenge-response protocol between it and the cloud at regular time.

The steps involved in key generation are listed below

1.The clients request the secret key from the TPA and the TPA sends the encrypted key to the client.

```
Key key = generateKey();
Cipher c = Cipher.getInstance(ALGO);
 c.init(Cipher.ENCRYPT_MODE, key);
```

2. Now, the client decrypts the secret key using AES decryption algorithm.

```
Key key = generateKey();
Cipher c = Cipher.getInstance(ALGO);
 c.init(Cipher.DECRYPT_MODE,    key);
```

The same key is used for Encryption and Decryption in AES Algorithm since it is a symmetric key Algorithm

Sub processes in encryption and decryption are ,

1.Initialize the state array with the block data (plaintext).

2.Adding 256 bit round keys which consists of 14 rounds.

3.Mixing columns and shifting rows

4.Then 16 input bytes are substituted by looking up the fixed table of rows and columns

B. UPLOAD DATA

The client sends the key requirement to the TPA only when he wants to upload new files to cloud. And then the TPA sends the encrypted secret key to the client. When the client wants to upload new files to the cloud, it needs to verify the validity of the encrypted secret key from the TPA and recover the real secret key. They only happen in the time periods when the client needs to upload new files to the cloud. Furthermore, verifying the correctness of the encrypted secret key is done by the TPA.
The clients uploads and downloads their text file using the secret key

```
FileItemFactory fif = new DiskFileItemFactory();
ServletFileUpload upload =
newServletFileUpload(fif);
```

C. AUDITING PROOF

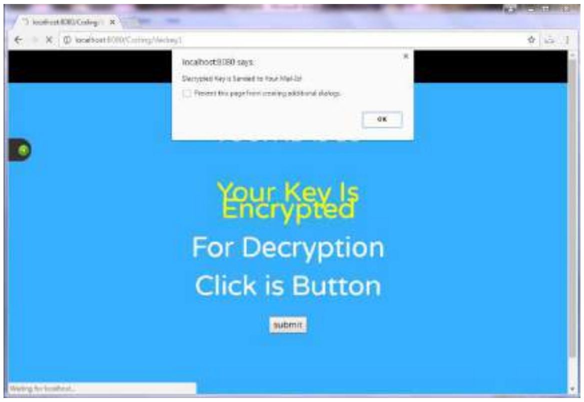
Outsource key updates for cloud storage auditing with key-exposure resilience. We propose the first cloud storage auditing protocol with verifiable outsourcing of key updates. In this protocol, key updates are outsourced to the TPA and are transparent for the client. The notion of “proof of retrievability” (PoR) was proposed to ensure both possession and retrievability of data at untrusted servers. Proposed a public privacy-preserving auditing protocol

D. KEY REQUIREMENTS AND RESULT

The client sends the key requirement to the TPA only when he wants to upload new files to cloud. And then the TPA sends the encrypted secret key to the client. After that, the client decrypts it to get his real secret key, generates authenticators for files, and uploads these files along with authenticators to cloud. In fact, shows the security should satisfy that an adversary cannot forge any authenticator in any time period even if it gets the decryption secret key *DK* by attacking the client.

IV. IMPLEMENTATION AND RESULTS

The proposed system is initialized with key request by client from the TPA. In this step the TPA provides the encrypted key to the client where the client can decrypt it. The key is generated randomly using AES key generation algorithm. Now the client can upload files and downloads it whenever necessary using the key. This method is developed using NetBeans IDE 7.3.1 version on windows platform. The result of the proposed method is shown in figure 2 to 4. In figure 4, shows the authority of Third Party Auditor to audit the users and their file and also the keys used by them. They also have the authority to remove unwanted files from the system.



localhost:8080 says:

Decrypted Key is Sended to Your Mail-Id

☐ Prevent this page from creating additional dialogs.

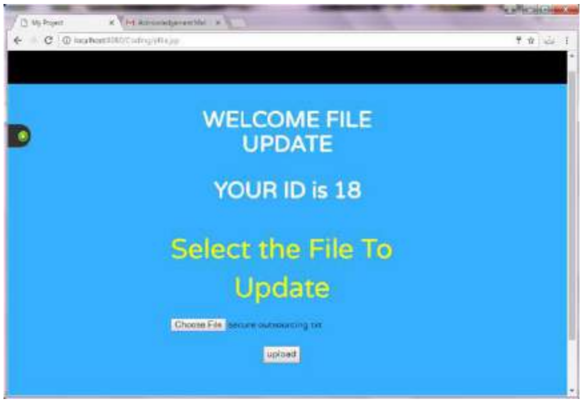
OK

Your Key Is Encrypted

For Decryption Click is Button

submit

Figure 2. Key Decryption



WELCOME FILE UPDATE

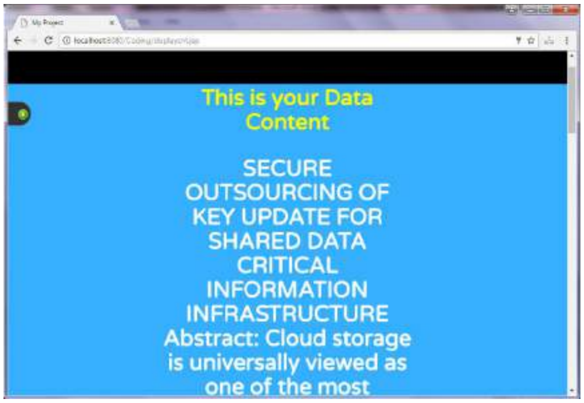
YOUR ID is 18

Select the File To Update

Choose File secure outsourcing.txt

upload

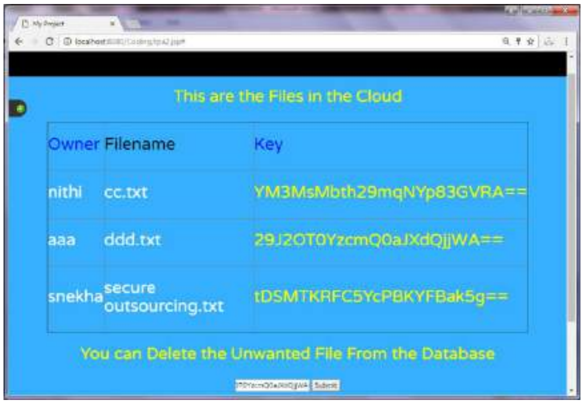
Figure 3. File upload



This is your Data Content

SECURE OUTSOURCING OF KEY UPDATE FOR SHARED DATA CRITICAL INFORMATION INFRASTRUCTURE Abstract: Cloud storage is universally viewed as one of the most

Figure 4. File display



This are the Files in the Cloud

Owner	Filename	Key
nithi	cc.txt	YM3MsMbth29mqNYp83GVRA==
aaa	ddd.txt	29J2OT0YzcmQ0aJXdQjjWA==
snekha	secure outsourcing.txt	tDSMTKRFC5YcPBKYFBak5g==

You can Delete the Unwanted File From the Database

Submit

Figure 5. Third Party Auditing

V. CONCLUSION

The proposed method enhances the security of the clients data and reduces the burden of the clients especially those with small computing devices such as mobile phones. Hence client can upload and download the files using the key provided by the Third Party Auditor (TPA).

ACKNOWLEDGEMENT

We would like to thank S. Sathea sree, Assistant professor, who inspired us and guided us to involve in research work. We would also like to thank Dr.A.Vijayakumar, Professor, Head of the Department, for continuous motivation and support. We are thankful to all those who helped us.

REFERENCE

1. Jia Yu, Kui Ren, *Fellow, IEEE*, and Cong Wang, *Member, IEEE, "*Enabling Cloud Storage Auditing With Verifiable Outsourcing of Key Updates", IEEE Transactions on Information Forensics and Security, Vol. 11, No. 6, June 2016.

2. Cong Wang, *Student Member, IEEE,* Sherman S.-M. Chow, Qian Wang, *Student Member, IEEE,*Kui Ren, *Member, IEEE,* and Wenjing Lou, *Member, IEEE, "*Privacy-Preserving Public Auditing for Secure Cloud Storage", IEEE Transactions on computers, Volume: 62, Issue: 2, Jan 2014.

3. C.C. Erway, A. Ku¨ pc¸u¨ , C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security, pp. 213-222, 2009.

4. G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth Int'l Conf. Security and Privacy in Comm. Netowrks (SecureComm), pp. 1- 10, 2008.

5. " Efficient Audit Service Outsourcing For Data Integrity in clouds", International journal on Engineering Technology and Sciences – IJETS ISSN (P): 2349-3968, ISSN (0): 2349-3976, Volume 1, Issue 7, November2014.